



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2003271445 A**(43) Date of publication of application: **26.09.03**

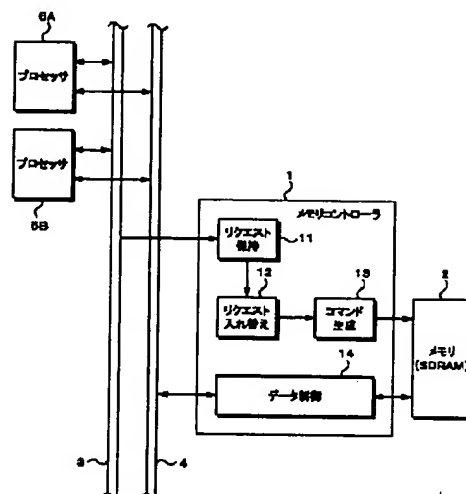
(51) Int. Cl.

G06F 12/00(21) Application number: **2002071321**(71) Applicant: **SONY CORP**(22) Date of filing: **15.03.02**(72) Inventor: **YAMANAKA KATSUHIKO****(54) MEMORY CONTROL DEVICE AND METHOD****(57) Abstract:**

PROBLEM TO BE SOLVED: To enhance access efficiency by causing read and write commands not to occur alternately when accessing to SDRAM.

SOLUTION: A request hold circuit 11 holds read requests and write requests from a processor 5A and 5B in order. In a request permutation circuit 12, the read requests or write requests held in the request hold circuit 12 are permuted so as to the read commands or the write commands are continued. In a command generation circuit 13, the command for accessing to a memory (SDRAM) 2 is generated. This prevents wasteful cycles from occurring to improve the access efficiency to the memory.

COPYRIGHT: (C)2003,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-271445

(P2003-271445A)

(43) 公開日 平成15年9月26日 (2003.9.26)

(51) Int.Cl.⁷

G 0 6 F 12/00

識別記号

5 6 0

5 9 7

F I

G 0 6 F 12/00

テ-リ-ト* (参考)

5 6 0 C 5 B 0 6 0

5 9 7 C

審査請求 未請求 請求項の数 8 O L (全 12 頁)

(21) 出願番号 特願2002-71321(P2002-71321)

(22) 出願日 平成14年3月15日 (2002.3.15)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 山中 克彦

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(74) 代理人 100082762

弁理士 杉浦 正知 (外1名)

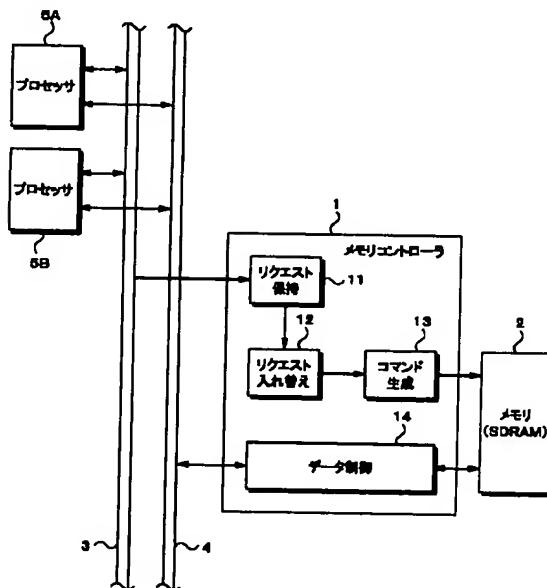
Fターム(参考) 5B060 CA04

(54) 【発明の名称】 メモリ制御装置及び方法

(57) 【要約】

【課題】 SDRAMをアクセスする際に、リードコマンドとライトコマンドとが交互に発生することがないようにして、アクセス効率を高めるようにする。

【解決手段】 リクエスト保持回路11には、プロセッサ5A、5Bからのリードリクエスト或いはライトリクエストが順に保持される。リクエスト入れ替え回路12で、リードコマンド同士或いはライトコマンド同士が続くように、リクエスト保持回路12に保持されているリードリクエスト或いはライトリクエストの順番が入れ替えられる。コマンド生成回路13でメモリ (SDRAM) 2をアクセスするためのコマンドが生成される。これにより、無駄なサイクルが発生することがなくなり、メモリのアクセス効率が向上する。



【特許請求の範囲】

【請求項1】 クロックを使用してデータのリード或いはライトを行う同期型メモリを制御するメモリ制御装置であって、

上記メモリに対するリードリクエスト或いはライトリクエストを保存するリクエスト保持手段と、

上記リクエスト保持手段に保持されているリクエストの中から、リードリクエスト同士或いはライトリクエスト同士が連続するようにリクエストの順番を入れ替えるリクエスト入れ替え手段と、

上記リクエストに応じて上記メモリをリード或いはライトするためのコマンドを生成するコマンド生成手段とを備えるようにしたメモリ制御装置。

【請求項2】 上記メモリは、SDRAMである請求項1に記載のメモリ制御装置。

【請求項3】 上記リクエスト入れ替え手段は、現在発行しているリクエストと、次に発行すべきリクエストとでリード同士或いはライト同士が続くかどうかを判断し、リード同士或いはライト同士が続かなければ、現在発行しているリクエストと、更にその次に発行すべきリクエストとでリード同士或いはライト同士が続くかどうかを判断し、現在発行しているリクエストと、更にその次に発行すべきリクエストとでリード同士或いはライト同士が続くなら、次に発行すべきリクエストと更に次に発行すべきリクエストとを入れ替えるようにした請求項1に記載のメモリ制御装置。

【請求項4】 上記リクエスト入れ替え手段は、更に、次に発行すべきリクエストと更に次に発行すべきリクエストとの入れ替えの回数をカウントするカウント手段を備え、上記次に発行すべきリクエストと更に次に発行すべきリクエストとの入れ替えの回数が所定値に達したら、次に発行すべきリクエストと更に次に発行すべきリクエストとの入れ替えを制限するようにした請求項3に記載のメモリ制御装置。

【請求項5】 クロックを使用してデータのリード或いはライトを行う同期型メモリを制御するメモリ制御方法であって、
上記メモリに対するリードリクエスト或いはライトリクエストを保存し、
上記保持されているリクエストの中から、リードリクエスト同士或いはライトリクエスト同士が連続するようにリクエストの順番を入れ替え、
上記リクエストに応じて上記メモリをリード或いはライトするためのコマンドを生成するようにしたメモリ制御方法。

【請求項6】 上記メモリは、SDRAMである請求項5に記載のメモリ制御方法。

【請求項7】 上記リクエスト入れ替えは、現在発行しているリクエストと、次に発行すべきリクエストとでリード同士或いはライト同士が続くかどうかを判断し、リ

ード同士或いはライト同士が続かなければ、現在発行しているリクエストと、更にその次に発行すべきリクエストとでリード同士或いはライト同士が続くかどうかを判断し、現在発行しているリクエストと、更にその次に発行すべきリクエストとでリード同士或いはライト同士が続くなら、次に発行すべきリクエストと更に次に発行すべきリクエストとを入れ替えるようにした請求項5に記載のメモリ制御方法。

【請求項8】 上記リクエスト入れ替えは、更に、次に発行すべきリクエストと更に次に発行すべきリクエストとの入れ替えの回数をカウントし、上記次に発行すべきリクエストと更に次に発行すべきリクエストとの入れ替えの回数が所定値に達したら、次に発行すべきリクエストと更に次に発行すべきリクエストとの入れ替えを制限するようにした請求項7に記載のメモリ制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、例えば、ビデオデータを処理するための画像処理チップで画像データを蓄えるメモリとして使用されるSDRAMの制御に用いて好適なメモリ制御装置及び方法に関するもので、特に、SDRAMのアクセス効率の向上に係わる。

【0002】

【従来の技術】例えば、MPEG (Moving Picture Coding Experts Group) 2のトランスポートストリームを入力し、このトランスポートストリームからビデオパケットやオーディオパケットを分離し、デコードされたビデオデータに対して、画像処理を行ったり、OSD (On Screen Display) 画像を重ねたりするような画像処理チップの開発が進められている。このような画像処理チップには、MPEG 2のTS (Transport Stream) パケットを分離するためのトランスポートストリームプロセッサや、種々の画像処理を行うためのグラフィックスエンジンや、任意のソース矩形領域の画像をデスティネーション領域にコピーするためのBitBLTエンジンや、OSD画面を生成して重ねたり、画面の拡大、縮小を行うディスプレイプロセッサ等が配される。これらのプロセッサの処理には、全て大容量のメモリが必要であり、各プロセッサでメモリが共用される。このような画像処理プロセッサのためのメモリとしては、SDRAM (Synchronous Dynamic Random Access Memory) が用いられている。

【0003】SDRAMは、クロックに同期した連続的なデータ転送が可能であり、バースト転送を指定すると、指定したバイト数分のデータ転送を1クロック単位で連続して行うことができる。また、SDRAM内の記憶領域は、複数のバンクに分割した構成とされている。

【0004】図8は、このような画像処理プロセッサで使用される従来のメモリコントローラのブロック図を示すものである。

【0005】図8において、メモリコントローラ101は、SDRAMからなるメモリ102に接続されており、メモリ102に対するデータの読み出し或いは書き込みの制御を行っている。メモリコントローラ101は、レジスタバス103及びメモリバス104を介して、各プロセッサ105A、105B、…に接続されている。プロセッサ105A、105B、…は、例えば、画像処理チップの場合には、トランスポートストリームプロセッサや、種々の画像処理を行うためのグラフィックスエンジン、BitBLT(Bit Block Transfer)エンジン、ディスプレイプロセッサ、ホストCPU(Central Processing Unit)等である。

【0006】メモリコントローラ101は、メモリ102に対するアクセスのリクエストを保持するリクエスト保持回路111と、リクエスト保持回路111からのリクエストをメモリ102へのコマンドとして発行するコマンド生成回路113と、リード/ライトされたデータの受け渡しを行うデータ制御回路114とからなる。

【0007】プロセッサ105A、105B、…は、メモリ102をアクセスする場合には、レジスタバス103を介して、メモリコントローラ101に対して、リクエストを送る。リクエストには、メモリ102からデータを読み出すリードリクエストと、メモリ102にデータを書き込むライトリクエストとがある。プロセッサ105A、105B、…からのリクエストは、リクエスト保持回路111に保持される。メモリ102から読み出されたデータは、メモリバス104を介して、メモリコントローラ101から各プロセッサ105A、105B、…送られる。また、メモリ102にデータを書き込む場合には、各プロセッサ105A、105B、…からメモリバス104を介して、メモリコントローラ101に書き込みデータが送られる。

【0008】リクエスト保持回路111には、プロセッサ105A、105B、…から送られてきたリクエストが順番に保持される。このリクエスト保持回路111に保持されているリクエストは、順番に、コマンド生成回路113に送られる。

【0009】コマンド生成回路113は、このリクエストに対応するメモリのコマンドを生成する。コマンドは、ある要求に対するバンクアクティブ、リード/ライト、プリチャージのコマンドセットからなる。コマンド生成回路113で生成されたコマンドは、メモリ102に与えられる。

【0010】コマンド生成回路113に送られたリクエストがリードリクエストの場合には、コマンド生成回路113から、読み出しのためのコマンドセットが発生される。これにより、メモリ102からデータが読み出される。読み出されたデータがデータ制御回路114に送られる。そして、この読み出されたデータは、メモリバス104を介して、リクエストを発行したプロセッサ1

05A、105B、…に送られる。

【0011】コマンド生成回路113に送られたリクエストがライトリクエストの場合には、コマンド生成回路113から書き込みのためのコマンドセットが発生される。リクエストを発行したプロセッサ105A、105B、…からは、メモリバス104を介して、データ制御回路114に、データが送られる。このデータ制御回路114に送られてきたデータがメモリ102に書き込まれる。

【0012】

【発明が解決しようとする課題】上述のように、従来のメモリコントローラ101では、リクエスト保持回路111に順番にリクエストが保持され、このリクエストが順番にコマンド生成回路113に送られる。コマンド生成回路113では、リクエスト保持回路111からのリクエストに基づいて、メモリ102を構成するSDRAMのコマンドセットが生成される。

【0013】SDRAMのコマンドには、アクティブ、リード/ライト、プリチャージがある。アクティブコマンドは、選択したメモリセルに対応するワード線に所定のパルス電圧(ワード線を活性化させる)ことを指示するコマンドである。リードコマンドは、選択されたメモリセルのデータ線の電位から記憶データを読み取ること指示するコマンドである。ライトコマンドは、選択されたメモリセルのデータ線に、書き込むデータに応じた電位を印加することを指示するコマンドである。プリチャージコマンドは、SDRAM内のアクセスを行うメモリセルに対応するデータ線を所定の電位に設定することを指示するコマンドである。

【0014】各プロセッサ105A、105B、…からのリクエストには、メモリ102からデータを読み出すためのリードリクエストと、メモリ102にデータを書き込むためのライトリクエストとがある。リクエスト保持回路111は、各プロセッサ105A、105B、…からのリクエストを順番に保持し、保持されているリクエストをコマンド生成回路113に順番に送っている。したがって、コマンド生成回路113から出力されるコマンドとしては、リードのコマンドが連続して出力される場合と、ライトのコマンドが連続して出力される場合と、リードのコマンドとライトのコマンドとが交互に出力される場合とが考えられる。

【0015】SDRAMでは、リードだけ、或いはライトだけを連続して行う場合には、リード或いはライトのコマンドを連続的に発行することができるため、データアクセスを連続的に行うことができ、データバスを有効に使用し効率を上げることができる。

【0016】ところが、リードとライトが交互に行われる場合には、リードコマンド及びライトコマンドと、リードコマンド及びライトコマンドに対応するデータとの間のレイテンシが異なることによるバスの衝突の回避等

に起因するSDRAMへのコマンド発行の制約により、リードデータとライトデータとの間に無効なサイクルが発生し、連続的にコマンドを発生することができない。このため、メモリのアクセス効率が低下するという問題が生じる。

【0017】上述の従来のメモリコントローラ101では、リードのコマンドセットとライトのコマンドとが交互に出力される場合があるため、メモリのアクセス効率の低下が生じる。

【0018】つまり、図9は、コマンド生成回路113 10にリードリクエストが連続的に送られてきた場合にコマンド生成回路113から発行されるコマンドと、メモリ102から入出力されるデータとの関係を示すタイミングチャートである。

【0019】なお、ここでは、リードコマンド1個に対し2個のデータが出力されるモード（バーストモード、バースト長2）とし、リードコマンドから対応する最初のデータが出力されるまでのレイテンシ（CASレイテンシ）が2クロックのSDRAMを想定している。また、簡単のために、全てのリードリクエストが同一バンクの同一カラムに対して発行されているものとしている。

【0020】図9において、時点T1でアクティブコマンド（ACT）が発行され、時点T3で、アドレスC1のリードコマンド（READ）が発行される。リードコマンドが連続する場合には、それから、時点T5、T7において、アドレスC2、C3のリードコマンド（READ）が連続的に発行される。なお、PREはプリチャージコマンドを示し、NOPはウェイトを示す。

【0021】バースト長が2、CASレイテンシが2クロックなので、時点T3でアドレスC1のリードコマンドが発行されると、それから2クロック後の時点T5 30で、アドレスC1のデータRd1が読み出され、時点T6で、アドレスC1のデータRd2が読み出される。

【0022】それから、時点T5、T7において、アドレスC2、C3のリードコマンドが連続的に発行されているので、これに対応して、時点T7及びT8でアドレスC2のデータRd3及びRd4が読み出され、時点T9及びT10でアドレスC3のデータRd5及びRd6 35が読み出される。

【0023】このように、リードリクエストが連続的に送られてきた場合には、時点T3、T5、T7でリードコマンドを連続的に発行することができ、時点T5からT10で、データRd1、Rd2、Rd3、Rd4、Rd5、Rd6が途切れることなく出力されることになる。

【0024】これに対して、図10は、コマンド生成回路113にリードリクエストとライトリクエストとが交互に送られてきた場合にコマンド生成回路113から発行されるコマンドと、メモリ102から入出力されるデ 50

ータとの関係を示すタイミングチャートである。

【0025】図10において、時点T1でアクティブコマンド（ACT）が発行される。そして、時点T3でアドレスC1のリードコマンド（READ）が発行される。

【0026】時点T3でアドレスC1のリードコマンドが発行されると、バースト長が2、CASレイテンシが2クロックなので、これに対応して、時点T5及びT6でアドレスC1のデータRd1及びRd2が読み出される。

【0027】それから、データの読み出しが終了した時点T7で、アドレスC2のライトコマンド（Write）が発行され、時点T7及び時点T8で、データWd1及びWd2が書き込まれる。

【0028】それから、データの書き込みが終了した時点T9で、アドレスC3のリードコマンド（READ）が発行され、これに対応して、時点T11及びT12でアドレスC3のデータRd3及びRd4が読み出される。

【0029】それから、データの読み出しが終了した時点T13で、アドレスC4のライトコマンド（Write）が発行され、時点T13及び時点T14で、データWd3及びWd4が書き込まれる。

【0030】このように、リードリクエストとライトリクエストとが交互に送られてきた場合には、リードコマンドに対するデータの読み出しが終了してから、ライトコマンドを出してデータを書き込み、ライトコマンドに対するデータの書き込みが終了してから、リードコマンドを出してデータの読み出しを行うことになる。この場合、時点T3において最初のリードコマンドを発行した後、次に続くライトコマンドの発行は、バスの衝突を防ぐため、時点T7で発行される。ライト後のリードコマンドの発行は時点T9に続くが、CASレイテンシのため、時点T9、T10においてデータバスが使用されないサイクルが発生し、効率を落とす原因となる。

【0031】したがって、この発明の目的は、リードコマンドとライトコマンドとが交互に発生することがないようにして、メモリのアクセス効率を高めるようにしたメモリ制御装置及び方法を提供することにある。

40 【0032】

【課題を解決するための手段】この発明は、クロックを使用してデータのリード或いはライトを行う同期型メモリを制御するメモリ制御装置であって、メモリに対するリードリクエスト或いはライトリクエストを保存するリクエスト保持手段と、リクエスト保持手段に保持されているリクエストの中から、リードリクエスト同士或いはライトリクエスト同士が連続するようにリクエストの順番を入れ替えるリクエスト入れ替え手段と、リクエストに応じてメモリをリード或いはライトするためのコマンドを生成するコマンド生成手段とを備えるようにしたメ

メモリ制御装置である。

【0033】この発明は、クロックを使用してデータのリード或いはライトを行う同期型メモリを制御するメモリ制御方法であって、メモリに対するリードリクエスト或いはライトリクエストを保存し、保持されているリクエストの中から、リードリクエスト同士或いはライトリクエスト同士が連続するようにリクエストの順番を入れ替え、リクエストに応じてメモリをリード或いはライトするためのコマンドを生成するようにしたメモリ制御方法である。

【0034】リクエスト保持回路には、リードリクエスト或いはライトリクエストが順に保持される。リクエストリードリクエストとライトリクエストとが交互に送られてきた場合には、リードコマンドに対するデータの読み出しが終了してから、ライトコマンドを出してデータを書き込み、ライトコマンドに対するデータの書き込みが終了してから、リードコマンドを出してデータの読み出しを行うことになる。ライト後のリードコマンドの発行は、CASレイテンシのため、データバスが使用されないサイクルが発生し、効率を落とす原因となる。

【0035】そこで、この発明では、リードコマンド同士或いはライトコマンド同士が続くように、リクエスト保持回路に保持されているリードリクエスト或いはライトリクエストの順番を入れ替えている。これにより、無駄なサイクルが発生することがなくなり、メモリのアクセス効率が向上する。

【0036】

【発明の実施の形態】以下、この発明の実施の形態について図面を参照して説明する。図1は、この発明の一実施の形態を示すものである。

【0037】図1において、メモリコントローラ1は、SDRAMからなるメモリ2に接続されており、メモリ2に対するデータの読み出し或いは書き込みの制御を行っている。メモリコントローラ1は、レジスタバス3及びメモリバス4を介して、各プロセッサ5A、5B、…に接続されている。プロセッサ5A、5B、…は、例えば、画像処理チップの場合には、トランスポートストリームプロセッサや、グラフィックスエンジン、BitBltエンジン、ディスプレイプロセッサ、ホストCPU等である。

【0038】メモリコントローラ1は、メモリ2に対するアクセスのリクエストを保持するリクエスト保持回路11と、リクエスト保持回路11に保持されているリクエストの中からリード或いはライトアクセスが連続するようにリクエストを選択してコマンド生成回路13に送るリクエスト入れ替え回路12と、リクエスト入れ替え回路12からのリクエストをメモリ2へのコマンドとして発行するコマンド生成回路13と、読み出し或いは書き込みされたデータの受け渡しを行うデータ制御回路14とからなる。

【0039】プロセッサ5A、5B、…は、メモリ2をアクセスする場合には、レジスタバス3を介して、メモリコントローラ1に対して、リクエストを送る。リクエストには、メモリ2からデータを読み出すリードリクエストと、メモリ2にデータを書き込みライトリクエストとがある。プロセッサ5A、5B、…からのリクエストは、リクエスト保持回路11に保持される。メモリ2から読み出されたデータは、メモリバス4を介して、メモリコントローラ1から各プロセッサ5A、5B、…送られる。また、メモリ2にデータを書き込み場合には、各プロセッサ5A、5B、…からメモリバス4を介して、メモリコントローラ1に送られる。

【0040】リクエスト保持回路11には、プロセッサ5A、5B、…から送られてきたリクエストが順番に保持される。このリクエスト保持回路11に保持されているリクエストは、リクエスト入れ替え回路12により、リード或いはライトアクセスが連続するように順番に入れ替わられて、コマンド生成回路13に送られる。

【0041】コマンド生成回路13は、このリクエストに対応するメモリのコマンドを生成する。このコマンドは、ある要求に対するバンクアクティブ、リード/ライト、プリチャージのコマンドセットからなる。コマンド生成回路13で生成されたコマンドは、メモリ2に与えられる。

【0042】コマンド生成回路13に送られたリクエストがリードリクエストの場合には、コマンド生成回路13から、読み出しのためのコマンドセットが発生される。これにより、メモリ2からデータが読み出される。読み出されたデータがデータ制御回路14に送られる。そして、この読み出されたデータは、メモリバス4を介して、リクエストを発行したプロセッサ5A、5B、…に送られる。

【0043】コマンド生成回路13に送られたリクエストがライトリクエストの場合には、コマンド生成回路13から書き込みのためのコマンドセットが発生される。このとき、リクエストを発行したプロセッサ5A、5B、…からは、メモリバス4を介して、データ制御回路14に、データが送られる。このデータ制御回路14に送られてきたデータがメモリ2に書き込まれる。

【0044】上述のように、この発明が適用されたメモリコントローラ1では、リクエスト入れ替え回路12が設けられ、このリクエスト入れ替え回路12により、リードアクセスとライトアクセスが連続して行われるように、リクエスト保持回路11からリクエストの順番に入れ替わられて、コマンド生成回路13に送られる。

【0045】なお、リクエスト入れ替え回路12は、後に説明するように、現在発行しているリクエストと、次に発行すべきリクエストとでリード同士或いはライト同士が続くかどうかを判断し、リード同士或いはライト同士が続かなければ、現在発行しているリクエストと、更

にその次に発行すべきリクエストとでリード同士或いはライト同士が続くかどうかを判断し、現在発行しているリクエストと、更にその次に発行すべきリクエストとでリード同士或いはライト同士が続くなら、次に発行すべきリクエストと更に次に発行すべきリクエストとを入れ替えるような処理を行う判断回路と、この判断回路で判断された順に、リクエスト保持回路11からのデータを選択して出力する選択回路とから構成されている。

【0046】このように、リクエスト入れ替え回路12により、リード同士或いはライト同士が続くようにリクエストが入れ替えられることにより、メモリアクセス効率の向上が実現される。

【0047】例えば、リクエスト保持回路11に、図2Aに示すように、リクエストreq0、req1、req2、req3が保持されているとする。そして、リクエストreq0はリード、リクエストreq1はライト、リクエストreq2はリード、リクエストreq3はライトの各リクエストであるとする。

【0048】このように、リードのリクエストとライトのリクエストとが交互に保持されている場合には、リクエスト入れ替え回路12により、リード或いはライト続くように、リクエストの順番が変更される。

【0049】すなわち、図2Aに示すように、リクエスト保持回路11に、リードのリクエストreq0、ライトのリクエストreq1、リードのリクエストreq2、ライトのリクエストreq3が保持されている場合には、図2Bに示すように、リクエスト入れ替え回路12により、リードのリクエストreq0、リードのリクエストreq2、ライトのリクエストreq1、ライトのリクエストreq3の順になるようにリクエストの順番が変更される。これにより、リード或いはライトのリクエストが続くようになる。

【0050】図3は、リクエスト保持回路11に保持されているリードのリクエストreq0、ライトのリクエストreq1、リードのリクエストreq2、ライトのリクエストreq3を、そのままの順にコマンド生成回路13に送った場合にコマンド生成回路13から発行されるコマンドと、メモリ2から入出力されるデータとの関係を示すタイミングチャートである。

【0051】なお、ここでは、リードコマンド1個に対し2個のデータが出力されるモード（バーストモード、バースト長2）とし、リードコマンドから対応する最初のデータが出力されるまでのレイテンシ（CASレイテンシ）が2クロックのSDRAMを想定している。また、簡単のために、全てのリードリクエストが同一バンクの同一カラムに対して発行されているものとしている。

【0052】図3において、時点T1でアクティブコマンド（ACT）が発行される。そして、時点T3でアドレスC1のリードコマンド（READ）が発行される。50

【0053】時点T3でアドレスC1のリードコマンドが発行されると、バースト長が2、CASレイテンシが2クロックなので、これに対応して、時点T5及びT6でアドレスC1のデータRd1及びRd2が読み出される。

【0054】それから、データの読み出しが終了した時点T7で、アドレスC2のライトコマンド（Write）が発行され、時点T7及び時点T8で、データWd1及びWd2が書き込まれる。

【0055】それから、データの書き込みが終了した時点T9で、アドレスC3のリードコマンド（READ）が発行され、これに対応して、時点T11及びT12でアドレスC3のデータRd3及びRd4が読み出される。

【0056】それから、データの読み出しが終了した時点T13で、アドレスC4のライトコマンド（Write）が発行され、時点T13及び時点T14で、データWd3及びWd4が書き込まれる。

【0057】図4は、リクエスト保持回路11に保持されているリードのリクエストreq0、ライトのリクエストreq1、リードのリクエストreq2、ライトのリクエストreq3を、リードのリクエストreq0、リードのリクエストreq2、ライトのリクエストreq1、ライトのリクエストreq3の順に入れ替えた場合にコマンド生成回路13から発行されるコマンドと、メモリ2から入出力されるデータとの関係を示すタイミングチャートである。

【0058】図4において、時点T1でアクティブコマンド（ACT）が発行される。そして、時点T3でアドレスC1のリードコマンド（READ）が発行され、時点T5でアドレスC2のリードコマンドが発行される。

【0059】バースト長が2、CASレイテンシが2クロックなので、時点T3で発行されたアドレスC1のリードコマンドに対応して、時点T5及びT6でアドレスC1のデータRd1及びRd2が読み出され、時点T5で発行されたアドレスC2のリードコマンドに対応して、時点T7及びT8でアドレスC1のデータRd3及びRd4が読み出される。

【0060】データの読み出しが終了した時点T9で、アドレスC3のライトコマンド（Write）が発行され、時点T11で、アドレスC4のライトコマンドが発行される。これに対応して、時点T9及び時点T10で、データWd1及びWd2が書き込まれ、時点T11及び時点T12で、データWd3及びWd4が書き込まれ、

【0061】このように、リードリクエストとライトリクエストとが交互に送られてきた場合には、リードコマンドに対するデータの読み出しが終了してから、ライトコマンドを出してデータを書き込み、ライトコマンドに対するデータの書き込みが終了してから、リードコマン

ドを出してデータの読み出しを行うことになる。この場合、時点T3において最初のリードコマンドを発行した後、次に続くライトコマンドの発行は、バスの衝突を防ぐため、時点T7で発行される。ライト後のリードコマンドの発行は時点T9に続くが、CASレイテンシのため、時点T9、T10においてデータバスが使用されないサイクルが発生し、効率を落とす原因となる。

【0062】これに対し、図4に示すように、2ワードのリードコマンド、2ワードのライトコマンドとなるように、コマンドの順番を変更した場合には、無駄なサイ

クルが発生することがなくなり、アクセス効率が上がる。

【0063】また、バスの衝突を防ぐために、最後のリードデータ時点と、最初のライトデータ時点との間を空けるために、ライトコマンドを1クロック遅らせるようにすることがある。すなわち、図3における時点T6と時点T7との間、時点T12と時点T13の間では、リードとライトが繋がっている。リードとライトとが繋がっていると、バス上で、メモリから読み出されるデータと、メモリに書き込むデータとが衝突を起こすことが考えられる。そこで、バスの衝突を防ぐために、最後のリードデータ時点と最初のライトデータ時点との間を空けるようにすることがある。

【0064】最後のリードデータ時点と最初のライトデータ時点との間を空けた場合、リクエスト保持回路11に保持されているリードのリクエストreq0、ライトのリクエストreq1、リードのリクエストreq2、ライトのリクエストreq3をそのままの順にコマンド生成回路13に送ると、図5に示すタイミングチャートのようになる。すなわち、図3におけるタイミングチャートの時点T6と時点T7との間、時点T12と時点T13との間では、リードデータとライトデータとが繋がっており、そこに、それぞれ、1クロックの遅延を入れたのが図5である。

【0065】これに対して、図6は、最後のリードデータ時間と、最初のライトデータ時間との間に1クロック遅延を入れ、リクエスト保持回路11に保持されているリードのリクエストreq0、ライトのリクエストreq1、リードのリクエストreq2、ライトのリクエストreq3を、リードのリクエストreq0、リードのリクエストreq2、ライトのリクエストreq1、ライトのリクエストreq3の順に入れ替えた場合のタイミングチャートである。すなわち、図4に示すタイミングチャートの時点T8と時点T9との間ではリードデータとライトデータとが繋がっており、そこに、1クロックの遅延を入れたのが図6である。

【0066】図3と図4を比較すると、図3の時点T9、T10でデータバスに無効なサイクルがあり、8ワード転送に10クロックかかっているが、図4では順番を入れ替えることによって、8ワードを8クロックで転

送できていることが分かる。

【0067】また、図5と図6を比較すると、図5では8ワード転送に12クロック必要としているが、図6では9クロックで終了している。したがって、バスの衝突を防ぐために、最後のリードと最初のライトとの間を空けるようにした場合には、この発明の効果がより大きくなる。

【0068】以上のように、リクエスト入れ替え回路12を付加し、リード或いはライトが連続になるようにリクエストの順番を入れ替えることにより、データアクセス効率を向上することができる。

【0069】なお、順番を入れ替える際の制限事項としては、同じユニットからのリクエストについては、順番を入れ替えないということがある。例えば、CPUがあるアドレスにデータを書き込み、それをすぐ読み込む場合、順番を入れ替えると結果が異なるので、リクエストにそのリクエストを出したユニットの情報を付加し、これを入れ替えの条件とすることで、このような場合の読み込みデータを保障する。

【0070】次に、このように、リード或いはライトが続くようにリクエストを入れ替えるリクエスト入れ替え回路12について説明する。

【0071】図7は、リクエスト入れ替え回路12の動作を示すフローチャートである。ここでは、リクエスト保持回路11に保持されるリクエストは、リクエストreq0、リクエストreq1、リクエストreq2であり、各リクエストreq0からreq3は、以下のよう

なものとする。

【0072】リクエストreq0は、現在処理中のリクエストの情報である。リクエストreq0がリードである場合は、リードが続くように、ライトである場合はライトが続くように選択を行う。

【0073】リクエストreq1は、現在、リクエスト保持回路11に保持されているリクエストの中で1番目のリクエストであり、次に発行すべきリクエストである。ただし、リクエスト入れ替え回路12によりリクエストreq2が先に選択された場合には、その後に発行されることになる。

【0074】リクエストreq2は、現在、リクエスト保持回路11に保持されているリクエストの中で2番目のリクエストであり、リクエストreq1の次に発行すべきリクエストである。ただし、リクエスト入れ替え回路12は、リクエストreq1より前にリクエストreq2を発行すべきかどうかを判定しており、リクエストreq2がリクエストreq1より先行して発行されることもある。

【0075】また、メモリコントローラ1に来るリクエストの順番によっては、あるリクエストがリクエストreq1として際限なく後回しとなる状態が起こり得る。これを防ぐために、リクエストreq1を後回しにする

ことができる回数の上限を外部から設定できるようにしている。この制限値を先発行閾値と呼ぶことにする。

【0076】また、セレクト信号は、次に処理すべきリクエストとして選択されたリクエストの番号を示す。

【0077】図7に示すフローチャートは、基本的に、リクエストReq 0が発行されたら、リクエストReq 1、リクエストReq 2の順にリクエストを発行していくが、リクエストReq 0とリクエストReq 1とでリードとリードが続く或いはライトとライトとが続かなければ、リクエストReq 0とリクエストReq 2とでリードとリードが続く或いはライトとライトとが続くかを判断し、リクエストReq 0とリクエストReq 2とでリードとリードが続く或いはライトとライトとが続くなら、リクエストReq 2をリクエストReq 1より先行して発行させることで、リード或いはライトが続くようにリクエストを入れ替えている。ただし、リクエストReq 1を後回しにすることができる回数の上限を先発行閾値内とるようにしている。

【0078】図7において、リクエストReq 2があるか否かが判断される(ステップS1)。ステップS1で、リクエストReq 2がなければ、次に発行すべきリクエストはリクエストReq 1になるので、先発行カウンタがリセットされ(ステップS2)、リクエストReq 1が発行される(ステップS3)。

【0079】ステップS1で、リクエストReq 2がある場合には、現在発行しているリクエストReq 0と次に発行すべきリクエストReq 1との関係が、リードが続く或いはライトが続く関係であるか否かが判断される(ステップS4)。

【0080】ステップS4で、リクエストReq 0とリクエストReq 1との関係が、リードが続く或いはライトが続く関係であるなら、次に発行すべきリクエストはリクエストReq 1になり、先発行カウンタがリセットされ(ステップS2)、リクエストReq 1が発行される(ステップS3)。

【0081】ステップS4で、リクエストReq 0とリクエストReq 1との関係が、リードが続く或いはライトが続く関係でなければ、現在発行しているリクエストReq 0と、リクエストReq 1の次に発行すべきリクエストReq 2との関係が、リードが続く或いはライトが続く関係であるか否かが判断される(ステップS5)

【0082】ステップS5で、リクエストReq 0とリクエストReq 2との関係が、リードが続く或いはライトが続く関係でないなら、リクエストReq 0とリクエストReq 1との関係も、リクエストReq 0とリクエストReq 2との関係も、共にリードが続く或いはライトが続く関係でないので、リクエストの入れ替えは行われない。この場合、次に発行すべきリクエストはリクエストReq 1になり、先発行カウンタがリセットされ

(ステップS2)、リクエストReq 1が発行される(ステップS3)。

【0083】ステップS5で、リクエストReq 0とリクエストReq 2との関係が、リードが続く或いはライトが続く関係であるなら、リクエストReq 1とリクエストReq 2とのアクセス領域が重なるか否かが判断される(ステップS6)。

【0084】リクエストReq 1がアクセスする領域と、リクエストReq 2がアクセスする領域が重なっている場合、順番を変えることはできない。この場合には、順番を入れ替えることなく、リクエストReq 1を発行することになり、先発行カウンタがリセットされ(ステップS2)、リクエストReq 1が発行される(ステップS3)。

【0085】なお、判定の方法は、リクエストReq 1、リクエストReq 2、それぞれのアクセス開始アドレス、終了アドレスを計算し、次の2つの条件

(1) リクエストReq 1の開始アドレス<リクエストReq 2の開始アドレス<リクエストReq 1の終了アドレス

(2) リクエストReq 1の開始アドレス<リクエストReq 2の終了アドレス<リクエストReq 1の終了アドレス

の両方を満たさない場合は、重なっていない、逆に、どちらか一方でも満たした場合は重なっているものと判定している。

【0086】ステップS6で、リクエストReq 1とリクエストReq 2とのアクセス領域が重ならない場合には、先発行カウンタの値が閾値以下か否かが判断され(ステップS7)、閾値以下なら、リクエストの入れ替えが行われる。この場合、次に発行すべきリクエストはリクエストReq 2になり、先発行カウンタがインクリメントされ(ステップS8)、リクエストReq 2が発行される(ステップS9)。

【0087】なお、この例では、リクエストReq 0とリクエストReq 1とでリードとリードが続く或いはライトとライトとが続かなければ、リクエストReq 0とリクエストReq 2とでリードとリードが続く或いはライトとライトとが続くかを判断し、リクエストReq 0とリクエストReq 2とでリードとリードが続く或いはライトとライトとが続くなら、リクエストReq 2をリクエストReq 1より先行して発行させることで、リード或いはライトが続くようにリクエストを入れ替えているが、リクエストの入れ替えの方法は、このような方法に限定されるものではない。

【0088】以上のように、この発明の実施の形態では、リクエスト保持回路11に保持されたリクエストを、そのままの順にコマンド生成回路13に送るのではなく、リード同士或いはライト同士が続くように、リクエストを入れ替えるようにしている。その結果、データ

アクセス効率を向上することができる。

【0089】

【発明の効果】この発明によれば、リードコマンド同士或いはライトコマンド同士が続くように、リクエスト保持回路に保持されているリードリクエスト或いはライトリクエストの順番を入れ替えるようにしている。これにより、無駄なサイクルが発生することがなくなり、メモリのアクセス効率が向上する。

【図面の簡単な説明】

【図1】この発明が適用されたメモリコントローラの一 10
例のブロック図である。

【図2】この発明が適用されたメモリコントローラの説明に用いる略線図である。

【図3】この発明が適用されたメモリコントローラの一
例の説明に用いるタイミングチャートである。

【図4】この発明が適用されたメモリコントローラの一
例の説明に用いるタイミングチャートである。 *

*【図5】この発明が適用されたメモリコントローラの一
例の説明に用いるタイミングチャートである。

【図6】この発明が適用されたメモリコントローラの一
例の説明に用いるタイミングチャートである。

【図7】リクエストの入れ替え処理の説明に用いるフロー
チャートである。

【図8】従来のメモリコントローラの一例のブロック図
である。

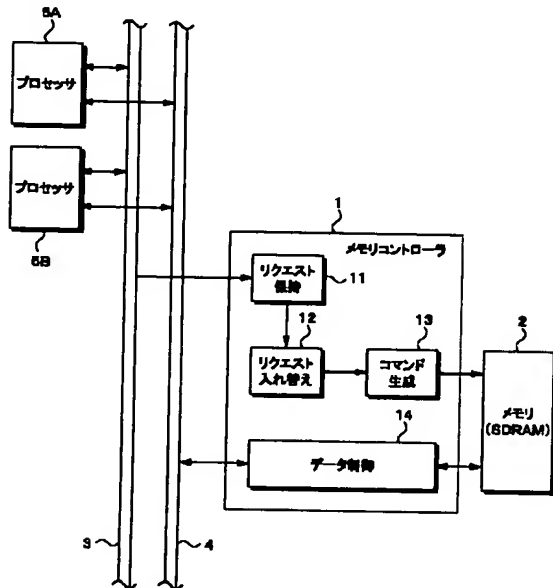
【図9】従来のメモリコントローラの一例の説明に用い
るタイミングチャートである。

【図10】従来のメモリコントローラの一例の説明に用
いるタイミングチャートである。

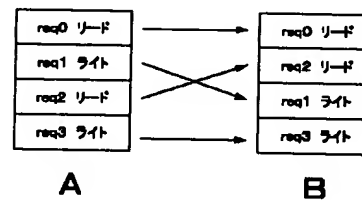
【符号の説明】

1・・・メモリコントローラ、2・・・メモリ、5A、
5B・・・プロセッサ、11・・・リクエスト保持回
路、12・・・リクエスト入れ替え回路、13・・・コ
マンド生成回路、14・・・データ制御回路

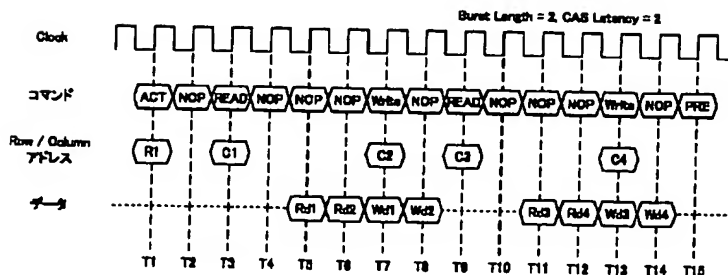
【図1】



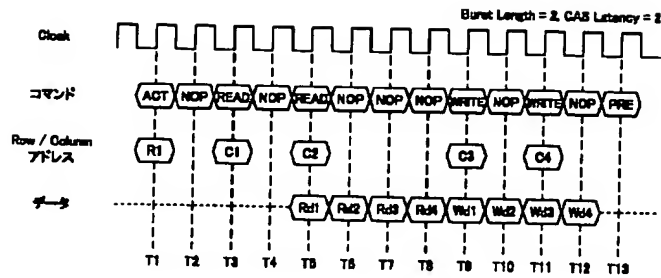
【図2】



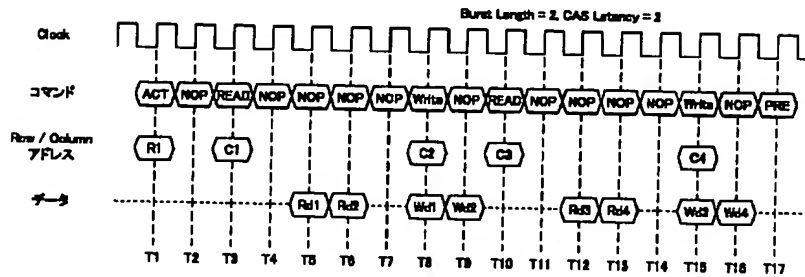
【図3】



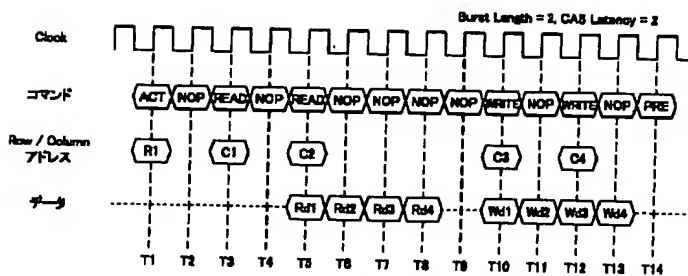
【図4】



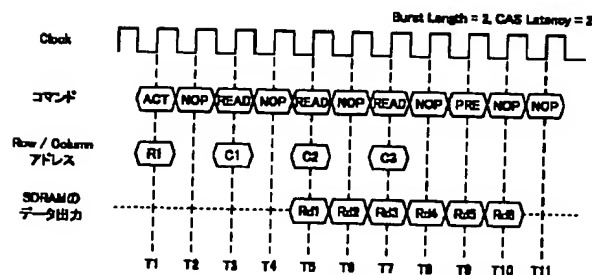
【図5】



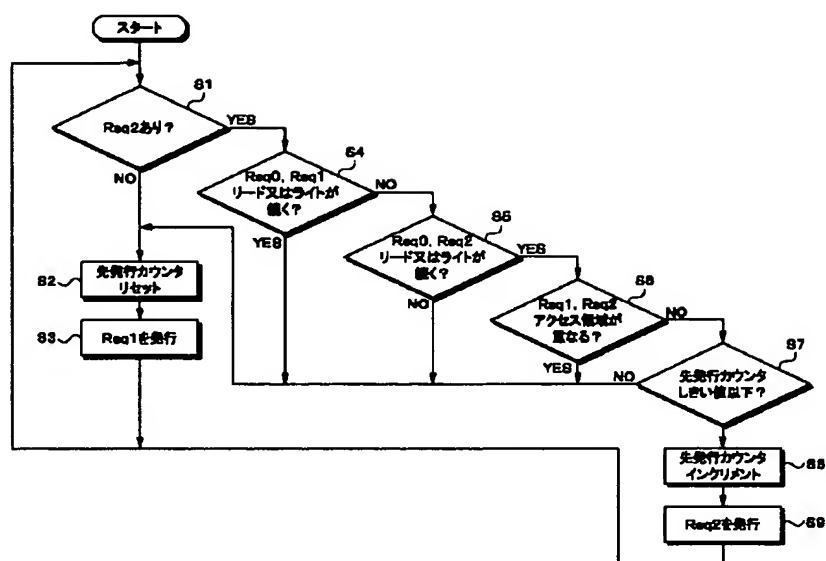
【図6】



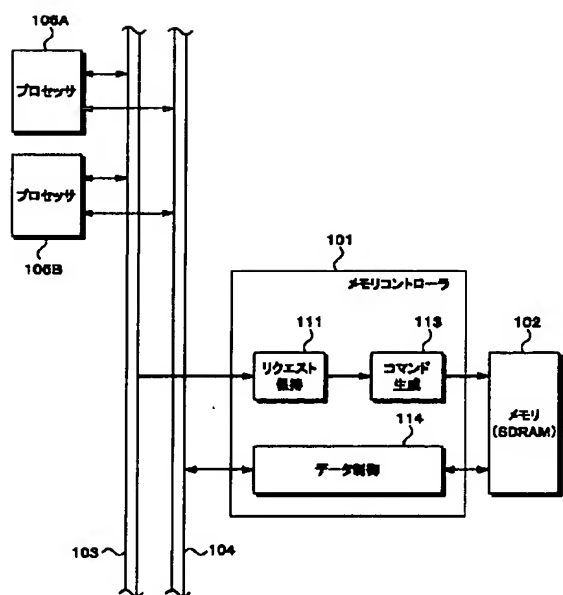
【図9】



【図7】



【図8】



【図10】

